# תרגיל 3

## React basics

* **נביאים**:
* <https://classroom.github.com/a/-KA7Ceij>

תאריך הגשה: 20/5/2024 23:59

* **שטראוס נשים**:
* <https://classroom.github.com/a/tbUtdGmd>

תאריך הגשה: 23/5/2024 23:59

* **שטראוס גברים**:
* <https://classroom.github.com/a/5UoOqaxi>

תאריך הגשה: 20/5/2024 23:59

בתרגיל זה תבנו דף אחד ב-react, הכולל משחק זיכרון למציאת זוגות תואמים.

בשלב ראשון הדף מציג קלט להזנת שם משתמש, כפתור להתחלת המשחק, כפתור להצגת ה-leaderboard בעזרת modal dialog או מסך נוסף עם כפתור חזרה, וכפתור הפותח (וסוגר) את הגדרות המשחק הבאות:

* + מספר שורות של לוח הקלפים (4 ברירת מחדל, מספר בין 2 ל-5)
  + מספר עמודות של לוח הקלפים ((4 ברירת מחדל, מספר בין 2 ל-5)
  + משך זמן (בשניות) תצוגת זוג קלפים (בין 0.5 ל-2.0 ). הספירה מתחילה רק אחרי הרמת הקלף השני.

Graphical user interface, text, application

Description automatically generated

פתיחת ה- settings:

Graphical user interface

Description automatically generated

תנאי התחלת המשחק:

* יש לבדוק שמספר הקלפים זוגי. כלומר אין לאפשר בחירה של 3x3, 5x5,3x5 . אין לאפשר התחלת משחק כל עוד הבחירה לא תקינה.  
    
  A screenshot of a memory game

  Description automatically generated
* חובה להזין שם (יש לנקות רווחים בהתחלה ובסוף ב-trim) באורך מקסימלי של 12 אותיות / ספרות בלבד. אין חשיבות לאותיות קטנות/גדולות כלומר Eli אותו שחקן כמו eli.

המשחק:

Table

Description automatically generated A picture containing application

Description automatically generated

אין לאפשר הרמת קלף כאשר נבחרו 2 קלפים לא מתאימים, כל עוד 2 הקלפים לא הוחזרו למצב תמונה כלפי מטה (בהתאם ל-delay של ה-settings).

המשחק מציג מספר קלפים (או זוגות) שהשחקן הרים, וכפתור ביטול המשחק וחזרה למסך כניסה.

חובה להשתמש ב-16 תמונות (באותו גודל) שישבו בפרויקט (ולא כתובות ברשת). יש להגריל את הקלפים בצורה רנדומלית כלומר בכל משחק תבחרו תמונות מתוך המאגר של 16 תמונות, ותבנו את לוח המשחק מחדש.

בסיום המשחק הדף מציג תוצאות:

* תוצאות השחקן: score, rank, מספר קלפים שהיו במשחק
* לוח התוצאות של כל השחקנים (leaderboard) עם מספור (rank), שם כל שחקן ו-score
* יש לעדכן את ה-leaderboard רק אם השחקן שיפר את ה-score שלו במידה והוא כבר מופיע ברשימה
* אופן חישוב ה-score: יש לפתח פונקציה של score על בסיס מספר קלפים של המשחק ומספר פעולות (אפשר גם להכניס את הזמן שלקח לסיים את משחק אם רוצים). יש לתעד את אופן חישוב הפונקציה ב-readme.

סיום משחק:

Table

Description automatically generated

תצוגת ה-leaderboard ע״י לחיצה על הכפתור במסך כניסה:

Table

Description automatically generated

שימו לב: אתם לא בונים משחק רשת (אחרת היינו צריכים backend). דמיינו שמשתמשים שונים משחקים מול אותו מחשב ושהתוצאות לוקאליות. טעינת הדף מאפסת את התכנית ולכן מאבדת את התוצאות המשחק בין הרצות. לכן אתם מתבקשים לאחסן תוצאות בדפדפן בעזרת [localstorage](https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage), דבר שיותר מתאים מאשר cookies ויאפשר לטעון את ה-leaderboard בתחילת התכנית. זו פעולה מאוד פשוטה: המרה ל-JSON ושמירה, וההפך:

localStorage.setItem('leaderboard', JSON.stringify(yourArray));

const leaderboard = JSON.parse(localStorage.getItem('leaderboard'));

חלק זה דורש לימוד useEffect של React דבר שנלמד בשבוע 5.

דרישות מבנה הפתרון:

* אין לאחסן את פתרון המשחק כך שניתן יהיה לפתוח את ה-inspector ,לחקור את ה-DOM ולגלות את הקלפים ההפוכים.
* יש להשתמש באלגוריתם מתאים לערבוב קלפים והכנת הלוח כגון [*Fisher–Yates*](https://en.wikipedia.org/wiki/Fisher–Yates_shuffle)על מנת להבטיח בחירה אקראית ומחודשת של הקלפים ועריכת הלוח. יש לתעד את האלגוריתם לבחירת הקלפים ב-readme.
* מבחינת החומר reactהפתרון יהיה מבוסס על states, מותר גם useEffect .מותר להשתמש בחומר הנלמד עד שבוע 5 של הסמסטר.

דרישות כלליות:

* דף רספונסיבי, שימוש ב-bootstrap
* הגדרת היררכיה של components ו-states מתאימה (להימנע מ-prop drilling מיותר, הגדרת יתר של states),שמירה על pure functions, תיעוד לכל פונקציה.
* שימוש ב-react-router למצבים (״דפים״) השונים
* ללא כפל קוד, ולהשתמש בכתיבה מודרנית (סגנון ES6)
* ניתן לעצב את הפתרון בצורה אחרת אך אין לחרוג משלבי המשחק ביחס לחווית המשתמש: זרימת בין מסכים (פתיחה, הגדרות, סיום משחק, leaderboard), פעולות קלט ותוכן. אם יש ספק, מומלץ להתייעץ.

בהצלחה!